Effective Soft-Adaptation for Neural Machine Translation

Shuangzhi Wu[1], Dongdong Zhang[2], and Ming Zhou[2]

[1] Harbin Institute of Technology, Harbin, China
[2] Microsoft Research Asia, Beijing, China
wushuangzhi2010@163.com, {dozhang,mingzhou}@microsoft.com

Abstract. Domain mismatch between training data and test data often degrades translation quality. It is necessary to make domain adaptation for machine translation tasks. In this paper, we propose a novel method to tackle Neural Machine Translation (NMT) domain adaptation issue, where a soft-domain adapter (SDA) is added in the encoder-decoder NMT framework. Our SDA automatically learns domain representations from the training corpus, and dynamically compute domain-aware context for inputs which can guide the decoder to generate domain-aware translations. Our method can softly leverage domain information to translate source sentences, which can not only improve the translation quality on specific domain but also be robust and scalable on different domains. Experiments on Chinese-English and English-French tasks show that our proposed method can significantly improve the translation quality of in-domain test sets, without performance sacrifice of out-of-domain/general-domain data sets.

Keywords: Domain adaptation · Machine translation.

1 Introduction

Neural machine translation (NMT) have been proved to be the current most effective approach to the machine translation task, which basically works in an attention-based encoder-decoder framework. Similar with the situation in statistical machine translation (SMT), NMT approach still suffers from the domain adaptation problem. Its performance degrades when domain mismatches between training data and test data. Motivated by previous work on domain adaptation for SMT, most existing work on NMT domain adaptation mainly focuses on either the data adaptation or the model adaptation.

Data adaptation can be performed either by data selection or instance weighting. For example, [18] revisited the instance weighting method in SMT and adapted it to NMT by assigning more weights on in-domain data and less weights on out-of-domain data in the objective function. This resulted in translation quality improvement on in-domain data but drop on out-of-domain data.

Model adaptation tries to transfer the model parameters to in-domain. In the work by [6,9,15] , NMT models were pre-trained over out-of-domain or general-domain data sets, followed by continuous training over in-domain corpus so that the model parameters can be fine-tuned towards in-domain test sets. To avoid the overfitting in the in-domain training stage, [5] proposed to train NMT models with the mixture of

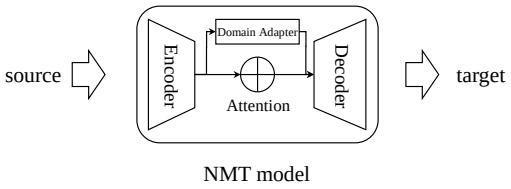

Fig. 1: Global overview of our model architecture.

pre-tagged training corpus where domain-specified tags were associated with source instances. They used oversampling to balance the proportion of in-domain and out-of-domain data. In the evaluation, the inputs also needed to be tagged into correct domains before performing translation.

These NMT domain adaptation approaches mostly concentrate on improving the effect of the small scale in-domain corpus when training NMT models. The more the in-domain training corpus affects the NMT model, the better performance could be achieved on in-domain testsets. They are a kind of static domain adaptation strategy, where prior domain types of inputs are needed before translating. In many practical translation tasks, the inputs always come from different domains and the domain information of each input may be not clearly described or hard to be captured. Previous methods are not robust and scalable enough to tackle cases like this. In addition, previous methods mostly focus on training models with translation quality improvement on in-domain test sets.

In this paper, we propose a dynamic domain adaptation approach for NMT, in which a novel soft-domain adapter (SDA) is introduced. Our SDA can learn domain representations from the training corpus. In decoding, SDA can dynamically generate domain context for each input sentences based on the encoder states and domain representations. As shown in Figure 1, on the basis of normal encoder-decoder NMT framework, there is an SDA consuming the encoder outputs to compute domain context of source inputs. Then, the decoder is fed with both the domain context and attention information to generate target translations. With SDA, our NMT decoder can softly leverage domain information to guide domain-aware translations. Therefore, our method belongs to a dynamic domain adaptation strategy, which not only can improve the translation quality of specific domain, but also can be scalable to multiple domains even without prior domain knowledge. We conducted the experiments on benchmark data sets of IWSLT2014 English-French and IWSLT2015 Chinese-English translation tasks. Experimental results show that our model significantly improves translation qualities on the in-domain test sets compared with NMT baselines and outperforms most of state-of-the-art methods.

2 NMT Background

NMT is an end-to-end framework [16,1] which directly models the conditional probability $P(Y|X)$ of target translation $Y = y_1,y_2,...,y_n$ given source sentence $X = x_1,x_2,...,x_m$,

where m and n are source and target length respectively. An NMT model consists of two parts: an encoder and a decoder. Both of them utilize recurrent neural networks (RNN) which can be a Long Short-Term Memory (LSTM) [7] or a Gated Recurrent Unit (GRU) [4] in practice. In this paper, we use GRU for all RNNs.

The RNN encoder bidirectionally encodes the source sentence into a sequence of context vectors $H = \boldsymbol{h}_1,\boldsymbol{h}_2,\boldsymbol{h}_3,...,\boldsymbol{h}_m$, where $\boldsymbol{h}_i = [\boldsymbol{h}_i, \boldsymbol{h}_i]$, $\boldsymbol{h}_i$ and $\boldsymbol{h}_i$ are calculated by two RNNs from left-to-right and right-to-left respectively. Then the decoder predicts target words one by one with probability

$$P(Y|X) = \prod_{j=1}^{n} P(y_j|y_{<j}, H) \tag{1}$$

Typically, for the jth target word, the probability $P(y_j|y_{<j}, H)$ is computed as

$$P(y_j|y_{<j}, H) = g(\boldsymbol{s}_j, y_{j-1}, \boldsymbol{c}_j) \tag{2}$$

where g is a nonlinear function that calculates the probability of y_j, and $\boldsymbol{s}_j$ is the RNN hidden state. The context $\boldsymbol{c}_j$ is calculated at each timestamp j based on H by the attention network [1]

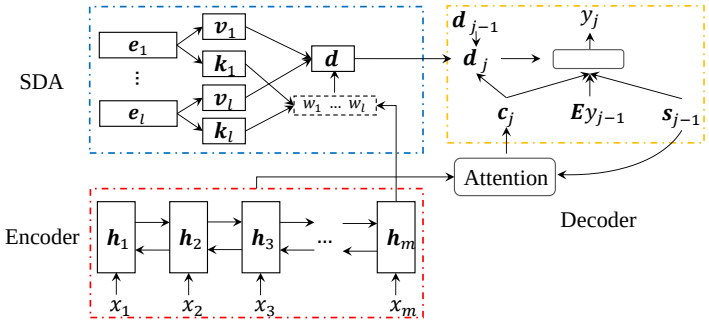

Fig. 2: Overview of our NMT model with soft-domain adapter (SDA-NMT). All characters in bold refers to vectors and Ey_i refers to embedding of y_i.

3 Our Method

In this paper, we propose to model domain distribution in the conventional NMT model. Given a source sentence $X = x_1,x_2,..,x_m$, its target translation $Y = y_1,y_2,..,y_n$, where m and n is source and target sentence length respectively, we define $D = D_1, D_2, .., D_l$ of l dimensions as the latent domain distribution of the source sentence where l is the number of different domains. D_i specifies the i-th dimension in D. We then introduce

this latent variable D into NMT. The original translation procedure of Equation 1 can be reformulated as

$$\begin{aligned} P(Y|X) &= \sum_{D_i \in D} P(Y, D|X) \\ &= \sum_{D_i \in D} P(Y|D, X) \cdot P(D|X) \\ &= \sum_{D_i \in D} P(y_1, y_2, .., y_n|D, X) \cdot P(D|X) \end{aligned} \tag{3}$$

For translation Y, it is generated as y_1,y_2,..,y_n following the way in a conventional sequence-to-sequence model. For domain distribution D, we design a novel **S**oft-**d**omain **A**dapter (**SDA**) for NMT to model domains of source sentences. Figure 2 sketches the high-level overview of our SDA-based NMT model where the SDA is added between the NMT encoder and decoder. The goal of SDA module is to learn domain representations from training data in the training phase. Then during decoding, SDA will generate a specialized domain context for the input sentence. The decoder will take the domain context as an extra input, further update it at each timestep and utilize it to generate domain-aware translation. Next we will describe the SDA module in detail.

3.1 Soft-domain Adapter (SDA) Module

For each domain D_i in D, we denote $\boldsymbol{e}_i$, $i \in [1, l]$, as its corresponding domain representation. [3] These representations are randomly initialized and trained during training as latent variables. Based on these representations, the SDA first maps $\boldsymbol{e}_i$ into two vectors by a linear transformation, denoted as $\boldsymbol{k}_i$ and $\boldsymbol{v}_i$,

$$\boldsymbol{k}_i = W_k \boldsymbol{e}_i + b_k \tag{4}$$

$$\boldsymbol{v}_i = W_v \boldsymbol{e}_i + b_v \tag{5}$$

where W_k, W_v are weight matrices and b_k and b_v are bias vectors. The $\boldsymbol{k}_i$ is used for indexing the corresponding domain D_i and $\boldsymbol{v}_i$ is used as value of D_i. Similar to the attention mechanism [1], we then calculate normalized similarity scores between the source sentence and each D_i by

$$w_i = \frac{\exp(\boldsymbol{h}_m^T \boldsymbol{k}_i)}{\sum_{j=1}^{l} \exp(\boldsymbol{h}_m^T \boldsymbol{k}_j)} \tag{6}$$

where $\boldsymbol{h}_m$ is the last concatenated hidden vector from the RNN encoder. In this way, we can get a normalized score vector, $\boldsymbol{W}$ = w_1,w_2,..,w_l. With this score vector, a specific domain context $\boldsymbol{d}$ for the current input can be generated by the following equations,

$$\widetilde{\boldsymbol{d}} = \sum_{i=1}^{l} w_i \boldsymbol{v}_i \tag{7}$$

$$\begin{aligned} \boldsymbol{d} &= \text{FFN}(\widetilde{\boldsymbol{d}}) \\ &= \max(0, \widetilde{\boldsymbol{d}} W_1 + b_1) W_2 + b_2 \end{aligned} \tag{8}$$

[3] In the rest of this paper, the characters in bold refer to vectors.

where W_1, W_2 are weight matrices and b_1, b_2 are biases. Equation 7 is a weighted sum operation. FFN (feed forward network) is a non-linear layer [17] which can be described as two convolutions with kernel size 1. We use FFN to extract more important features in $\widetilde{\boldsymbol{d}}$. $\boldsymbol{d}$ is then used in the decoder. The top-left part of Figure 2 gives a brief description of SDA module. Due to space limitation, the detailed decoding procedure is only illustrated at timestamp j. Our SDA is added between the NMT encoder and decoder. The encoder part and attention mechanism are the same with a conventional NMT model as described in Section 2. Next we will introduce our domain-aware decoder.

3.2 Domain-aware Decoder

We incorporate the domain context into decoder, the Equation 2 is rewritten as below,

$$P(y_j|y_{<j}, D, H) = g(\boldsymbol{s}_j, y_{j-1}, \boldsymbol{c}_j, \boldsymbol{d}_j) \tag{9}$$

$\boldsymbol{d}_j$ is updated at each decoding step j based on $\boldsymbol{d}$, $\boldsymbol{c}_j$ and $\boldsymbol{d}_{j-1}$.

The reason we update the domain context $\boldsymbol{d}$ at each timestep is motivated by the observation that: mostly, some parts of the source sentence may be domain sensitive, while other parts may be domain in-sensitive such as some function words, common words. In decoder, we update $\boldsymbol{d}_j$ by the following equation,

$$\boldsymbol{d}_j = \boldsymbol{r}_j \otimes \boldsymbol{d} \tag{10}$$

where $\boldsymbol{r}_j$ is an update gate formulated by

$$\boldsymbol{r}_j = \sigma(W_r \boldsymbol{c}_j + U_r \boldsymbol{d}_{j-1}) \tag{11}$$

where W_r, U_r are weight matrices, $\boldsymbol{c}_j$ is the source context calculated by the attention mechanism, σ is the sigmoid activation function and $\otimes$ is the element-wise multiplication. The right part of Figure 2 gives a brief description of our DA-based decoder. The attention mechanism follows the standard structure as described in [1].

3.3 Model Training

Our NMT model is trained on the mixture of multi-domain corpus. To ensure our SDA module can accurately learn domain representations, we propose an extra domain-related objective function to guide the model training. In our translation task, we can acquire the domain tag of each training instance (i.e. which sentence belongs to in-domain and which is from out-of-domain) in advance. For each training instance, we define a golden one-hot domain vector $\boldsymbol{G}$ with l dimension, l is the number of domains. We calculate another cross-entropy loss between the domain weights $\boldsymbol{W}$ described in Section 3.1 and the golden vector $\boldsymbol{G}$,

$$J^D(\theta) = \sum_{(X,D)\in S} \log\ P(D|X) \tag{12}$$

where S is the training set, θ is the model parameters. Then we add this function to the original cross-entropy loss to form the final objective function,

$$J(\theta) = \sum_{(X,Y,D)\in S} \log\ P(Y|X,D) + \log\ P(D|X) \tag{13}$$

With this function, our SDA module is trained in a supervised way. We can also train our model without $\log\ P(D|X)$. Thus the SDA module is implicitly trained and the objective function is the same with the conventional NMT model. We will further discuss the effect of term $\log\ P(D|X)$ in experiments. In the following parts of this paper, we use **SDA-NMT-DG** to represent the domain guided SDA model and **SDA-NMT** to represent the SDA-NMT-DG without the domain objective function $\log\ P(D|X)$.

4 Experiments

4.1 DataSet

In the Chinese-English task, we leverage the high quality bilingual data from IWSLT 2015 workshop [2] which contains about 200K sentence pairs as in-domain corpus. We use the dev 2010 set for development and test 2010-2015(tst2010-tst2015) are used as in-domain testsets. For out-of-domain corpus, we use a subset from LDC corpus [4] which has around 2.6M sentence pairs from News domain. NIST 2003, NIST 2005, NIST 2006, NIST 2008 and NIST 2012 are used as out-of-domain testsets. All English words are lowercased.

In the English-French translation task, the IWSLT 2014 English-French training corpus [3] is used as in-domain training data and the out-of-domain corpus is from WMT 2015 English-French translation task. The development data is TED dev2010 and we use test 2010 (tst2010) as testset. Both are with single reference per source sentence.

	tst2010	tst2011	tst2012	tst2013	tst2014	tst2015	Average
NMT-IWSLT (in-domain)	12.29	16.18	14.30	15.05	12.15	14.91	14.15
NMT-LDC (out-of-domain)	10.54	13.51	12.09	13.91	12.13	14.77	12.83
Fine-tuning	14.27	17.96	15.11	16.39	14.64	16.56	15.83
[5] (Mixed fine-tuning)	14.73	18.83	16.21	17.50	15.62	17.82	16.80
[13] (+Discriminator)	14.89	19.18	16.38	18.09	15.43	19.10	17.18
SDA-NMT	14.78	19.35	16.40	18.26	15.73	19.04	17.26
SDA-NMT-DG	**15.42**	**20.04**	**17.28**	**19.50**	**16.32**	**19.95**	**18.09**

Table 1: Evaluation results of Chinese-English in-domain test sets with BLEU% metric. "NMT-IWSLT" refers to a conventional NMT model trained on in-domain corpus and "NMT-LDC" denotes an NMT model trained on out-of-domain data.

[4] LDC2002E17, LDC2002E18, LDC2003E07, LDC2003E14, LDC2005E83, LDC2005T06, LDC2005T10, LDC2006E17, LDC2006E26, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006T06, LDC2004T08, LDC2005T10

4.2 Implementation Details

As we have just an in-domain and an out-of-domain, the hyper parameter l is set to 2. In the neural network training, the vocabulary size is limited to 35K high frequent words for both source and target languages in the Chinese-English translation task. All low frequent words are normalized into a special token `unk` and post-processed by following the work in [11]. For English-French task, we further split the words into sub-words using byte pair encoding (BPE)[5] [14] which has been shown to be effective for rare word problem in NMT. For all the NMT models, the size of word embedding is set to 512. The dimensions of the hidden states for all RNNs are set to 1024. The dimension of $\boldsymbol{k}_i$ and $\boldsymbol{v}_i$ are set to 512. The inner layer of FNN is set to 1024.

4.3 Baselines

We compare our proposed method with original NMT baselines and several state-of-the-art domain adaptation methods.

- NMT : An in-house reimplementation of [1]. In the following of this paper, we use NMT-IWSLT to represent the NMT model trained on in-domain corpus, NMT-WMT and NMT-LDC refers to NMT model trained on out-of-domain data.
- Fine-tuning : Fine-tune the out-of-domain model on the in-domain corpus.
- Mixed fine-tuning : [5] proposed a new training procedure named mixed fine-tuning.
- Instance weighting : [18] proposed to assign different weights to in-domain and out-of-domain instances.
- +Discriminator : [13] proposed a multi-task learning framework for NMT domain adaptation.

All the evaluation results are reported with the case-insensitive IBM BLEU-4 [12].

4.4 Evaluation on IWSLT In-domain Chinese-English Task

We first evaluate our method on IWSLT In-domain Chinese-English translation task. The evaluation results of in-domain testsets against baselines are listed in Table 1. We can see that NMT-IWSLT, which is trained on small scale corpus, is much better than NMT-LDC on the in-domain testsets in terms of average BLEU score. That is mainly because the domain of IWSLT corpus (spoken domain) is different from LDC corpus (news domain) and NMT models are sensitive to the training domain.

Though simple, "Fine-tuning" is an effective method which can improve the performance a lot compared with both NMT-IWSLT and NMT-LDC. "Mixed fine-tuning [5]" and "+Discriminator [13]" can achieve further improvements compared with "Fine-tuning". Overall, our SDA-NMT-DG achieves the highest BLEU scores on all the testsets. Compared with NMT-IWSLT baseline, our SDA-NMT-DG gains 3.94 more BLEU points on average. This is mainly because our method can generate dynamically domain-aware context for each instance where proper domain knowledge can be taken into

[5] https://github.com/rsennrich/subword-nmt

account during decoding. We also investigate the effect of domain objective function. Even without this objective function, the SDA-NMT can still outperform other methods on most of the testsets, which shows that our method can implicitly learn domain knowledge. However, the SDA-NMT is not as good as SDA-NMT-DG, this demonstrates that the golden domain tags can benefit the SDA module in training.

4.5 Evaluation on In-domain IWSLT English-French Task

In this section, we further evaluate our method on IWSLT English-French translation task. Table 2 shows the comparison results from 7 systems with the evaluation metrics of BLEU. A state-of-the-art result taken from [18] on this testset is also listed which proposed a "Instance weighting" strategy for NMT.

According to Table 2, our SDA-NMT-DG still outperforms the other models, where about 7 more BLEU points are gained compared to NMT-IWSLT baseline. Compared with the [18], SDA-NMT-DG achieves 1 more BLEU score. This shows that our proposed approach to modeling domain-aware context benefits NMT systems on in-domain testsets. In addition, the SDA-NMT-DG is still better than SDA-NMT.

	dev2010	tst2010
NMT-IWSLT (in-domain)	25.25	30.57
NMT-WMT (out-of-domain)	25.42	29.73
[18] (Instance weighting)	30.40	36.50
[5] (Mixed fine-tuning)	30.88	36.66
[13] (+Discriminator)	31.18	36.87
SDA-NMT	31.59	36.96
SDA-NMT-DG	**31.86**	**37.54**

Table 2: Evaluation results of English-French IWSLT dev and test sets with BLEU% metric.

4.6 Evaluation on Out-of-domain NIST Chinese-English Task

In this section, we investigate the performance of SDA-NMT-DG on the NIST out-of-domain Chinese-English translation task. Table 3 shows all the evaluation results. From the table, we can see that "Fine-tuning" performs the worst with more than 11 BLEU scores decrease compared with NMT-LDC in terms of average BLEU. This shows that even though "Fine-tuning" can improve the in-domain performance, it dramatically deteriorates the out-of-domain translation quality.

Overall, our SDA-NMT-DG achieves comparable BLEU scores compared with NMT-LDC baseline and is better than all the other domain adaptation systems. This is because our SDA can dynamically generate domain-aware representations and guide the decoder to generate out-of-domain translations. We can also find that, for some tests such as NIST2006, SDA-NMT-DG is even better than NMT-LDC baseline. Actually, these testsets are multilingual sets where Web data is contained. These web data may

be drawn from user forums, discussion groups, and blogs. They are more likely to the TED data. For these multiple domain testsets, our model can perform even better than NMT-LDC.

	NIST2003	NIST2005	NIST2006	NIST2008	NIST2012	Average
NMT-LDC (out-of-domain)	**41.85**	**39.58**	39.96	30.49	29.80	36.34
Fine-tuning	29.17	28.11	26.09	21.31	20.60	25.06
[5] (Mixed fine-tuning)	39.23	37.94	36.98	28.38	27.10	33.97
[13] (+Discriminator)	39.36	37.38	38.59	28.81	28.36	34.50
SDA-NMT	41.03	38.97	38.81	30.50	29.00	35.66
SDA-NMT-DG	41.10	39.20	**40.64**	**31.14**	**30.08**	**36.43**

Table 3: Evaluation results of Chinese-English out-of-domain test sets with BLEU% metric.

5 Related Work

Recently, neural machine translation (NMT) has achieved better performance than SMT in many language pairs [10,19]. Our work builds on the recent literature on domain adaptation strategies in NMT. The NMT model is trained in an end-to-end way which is very sensitive to the training domain. Some effort has been done to improve the NMT model on the in-domain testsets. [9] proposed to transfer out-of-domain knowledge to in-domain by fine-tuning out-of-domain models on in-domain corpus. [8] involved appending a domain indicator token to each source sequence. Based on these work, [5] further refined the model by integrating source-tokenization into the domain fine-tuning paradigm. While it requires no changes to the NMT architecture, these approaches are inherently limited because they stipulate that domain information for unseen test examples be known. For example, if using a trained model to translate user-generated sentences, we do not know the domain a-prior, and this approach cannot be used. There is another line of work. Inspired by the instance weighting methods in SMT, [18] applied this method to NMT models by assign different weights for each instance during training. More weights are assigned to in-domain instances in the NMT loss function.

Different from the work above, [13] proposed a multi-task learning framework for domain adaptation. They added a discriminator on the top of the NMT encoder which is used to classify which domain the source sentence belongs to. However, this is not effective enough to leverage domain information for NMT model. In this paper, we introduce SDA into the conventional NMT model. This module can learn domain distribution of training data and generate domain-aware representation for each source instances.

6 Conclusion and Future Work

In this paper, we propose a novel soft-domain adapter based neural machine translation model. Our model can learn domain representations from the training data and gener-

ate domain-aware context for input sentences. Then the decoder can generate domain-aware translations with the help of domain contexts. Experimental results show that our method can boost the translation quality on the in-domain testsets without deteriorating the out-of-domain performance.

In future work, along this research direction, we will conduct multiple domain translation experiments (more than two), such as a mixture domain of spoken, news and travel, to verify the effectiveness.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. ICLR 2015 (2015)
2. Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., Federico, M.: The iwslt 2015 evaluation campaign. Proc. of IWSLT, Da Nang, Vietnam (2015)
3. Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Federico, M.: Report on the 11th iwslt evaluation campaign, iwslt 2014. In: Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam (2014)
4. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of ENMLP 2014
5. Chu, C., Dabre, R., Kurohashi, S.: An empirical comparison of simple domain adaptation methods for neural machine translation. arXiv preprint arXiv:1701.03214 (2017)
6. Freitag, M., Al-Onaizan, Y.: Fast domain adaptation for neural machine translation. arXiv preprint arXiv:1612.06897 (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997)
8. Kobus, C., Crego, J., Senellart, J.: Domain control for neural machine translation. arXiv preprint arXiv:1612.06140
9. Luong, M.T., Manning, C.D.: Stanford neural machine translation systems for spoken language domains. In: Proceedings of IWSLT2015
10. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of EMNLP 2015
11. Luong, T., Sutskever, I., Le, Q., Vinyals, O., Zaremba, W.: Addressing the rare word problem in neural machine translation. In: Proceedings of ACL2015
12. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of ACL 2002 (2002)
13. Pryzant, R., Britz, D., Le, Q.: Effective domain mixing for neural machine translation. In: Second Conference on Machine Translation (WMT) (2017)
14. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
15. Servan, C., Crego, J., Senellart, J.: Domain specialization: a post-training domain adaptation for neural machine translation. arXiv preprint arXiv:1612.06141 (2016)
16. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems (2014)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. Curran Associates, Inc. (2017)
18. Wang, R., Utiyama, M., Liu, L., Chen, K., Sumita, E.: Instance weighting for neural machine translation domain adaptation. In: Proceedings of EMNLP2017
19. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)